35.C15104 2185

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#4

In re Application of:	)	
	:	
MASANARI TODA	)	
	:	Group Art Unit: 2185
Application No.: 09/779,642	)	
	:	
Filed: February 9, 2001	)	
	:	
For: INFORMATION PROCESSING APPARATUS, INFORMATION PROCESSING METHOD, AND MEMORY MEDIUM STORING PRINT CONTROL PROGRAM THEREIN	)	
	:	May 14, 2001

Commissioner for Patents
Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicant hereby claims priority under the International Convention and all rights to which he is entitled under 35 U.S.C. § 119 based upon Japanese Priority Application No. 2000-033757, filed February 10, 2000.

A certified copy of the priority document is enclosed.

Applicant's undersigned attorney may be reached in our Costa Mesa office by telephone at (714) 540-8700. All correspondence should continue to be directed to our address given below.

Respectfully submitted,

Attorney for Applicant

Registration No. 36,174

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

CFO 15104 US /shi

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年 ２月１０日

出願番号
Application Number: 特願２０００－０３３７５７

出願人
Applicant (s): キヤノン株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

２００１年 ３月 ２日

特許庁長官
Commissioner,
Patent Office 及川耕造

【書類名】　特許願
【整理番号】　4055028
【提出日】　平成12年 2月10日
【あて先】　特許庁長官　殿
【国際特許分類】　G06F 13/00
【発明の名称】　情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体
【請求項の数】　20
【発明者】
【住所又は居所】　東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内
【氏名】　戸田　雅成
【特許出願人】
【識別番号】　000001007
【氏名又は名称】　キヤノン株式会社
【代理人】
【識別番号】　100077481
【弁理士】
【氏名又は名称】　谷　義一
【選任した代理人】
【識別番号】　100088915
【弁理士】
【氏名又は名称】　阿部　和夫
【手数料の表示】
【予納台帳番号】　013424
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1

【物件名】　　要約書　1

【包括委任状番号】　9703598

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体

【特許請求の範囲】

【請求項１】　入力された印刷指令情報を格納する記憶手段と、

前記記憶手段に前記印刷指令情報を格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記記憶手段に格納させる合成手段と

を具備したことを特徴とする情報処理装置。

【請求項２】　請求項１に記載の情報処理装置において、

前記合成手段により、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記記憶手段に格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記記憶記憶手段に格納することを特徴とする情報処理装置。

【請求項３】　請求項１に記載の情報処理装置において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする情報処理装置。

【請求項４】　請求項３に記載の情報処理装置において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする情報処理装置。

【請求項５】　請求項１に記載の情報処理装置において、

前記合成手段は、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする情報処理装置。

【請求項６】　請求項１に記載の情報処理装置において、

前記合成手段は、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求める手段と、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップする手段とを

備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記記憶手段に格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記記憶手段に格納することを特徴とする情報処理装置。

【請求項７】　入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを有することを特徴とする情報処理方法。

【請求項８】　請求項７に記載の情報処理方法において、

前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納することを特徴とする情報処理方法。

【請求項９】　請求項７に記載の情報処理方法において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする情報処理方法。

【請求項１０】　請求項９に記載の情報処理方法において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする情報処理方法。

【請求項１１】　請求項７に記載の情報処理方法において、

前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする情報処理方法。

【請求項１２】　請求項７に記載の情報処理方法において、

前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップ

とを備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納することを特徴とする情報処理方法。

【請求項１３】　入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを、読み出し可能なプログラムの形態で記憶したことを特徴とする記憶媒体。

【請求項１４】　請求項１３に記載の記憶媒体において、

前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納することを特徴とする記憶媒体。

【請求項１５】　請求項１３に記載の記憶媒体において、

前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行うことを特徴とする記憶媒体。

【請求項１６】　請求項１５に記載の記憶媒体において、

前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送されることを特徴とする記憶媒体。

【請求項１７】　請求項１３に記載の記憶媒体において、

前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行うことを特徴とする記憶媒体。

【請求項１８】　請求項１３に記載の記憶媒体において、

前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、

前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップ

とを備え、

前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納することを特徴とする記憶媒体。

【請求項１９】　請求項１３～１８のいずれかに記載の記憶媒体において、

前記記憶媒体として、サーバ・コンピュータおよびクライアント・コンピュータが読むことができるプログラムを格納したフロッピーディスク、ハードディスク、光磁気ディスク、光ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭを用いることを特徴とする記憶媒体。

【請求項２０】　請求項１３～１８のいずれかに記載の記憶媒体において、

前記記憶媒体は、サーバ・コンピュータおよびクライアント・コンピュータに着脱可能であることを特徴とする記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体に関するものである。

【０００２】

更に詳述すると、本発明は、例えば所定の通信媒体を介して印刷装置を制御するのに好適な、ホストコンピュータとして機能する情報処理装置および情報処理方法およびプリンタドライバプログラムが格納された記憶媒体に関するものである。

【０００３】

【従来の技術】

まず、従来の印刷システムにおける情報処理方法および処理の流れについて、図２を用いて説明する。

【０００４】

従来の印刷システムでは、一般には、図２に示すように、セントロニクスイン

ターフェースといったパラレル通信手段やネットワーク通信手段を介してホストコンピュータとレーザービームプリンタとが接続されている。

【０００５】

ホストコンピュータ側では、ワードプロセッサや表計算のようなアプリケーションソフトウェア２０１がウインドウズ（Windows ：米国Microsoft 社の登録商標）のようないわゆる基本ソフトの上で動作している。こうしたアプリケーションに基づいて印刷を行う場合は、基本ソフトが提供するいくつかのサブシステムのうちグラフィックサブシステム２０２の機能が用いられる。このグラフィックサブシステムは、例えばWindows では、ＧＤＩ（Graphic Device Interface）２０２１と呼ばれており、ディスプレイあるいはプリンタに対する画像情報の処理を司っている。このＧＤＩ２０２１は、ディスプレイあるいはプリンタといった各デバイス毎の依存性を吸収するためにデバイスドライバと呼ばれるモジュールを動的にリンクし、それぞれのデバイスに対する出力処理を行う。プリンタに対するこのモジュールは、プリンタドライバ２０２３と呼ばれる。

【０００６】

このプリンタドライバ２０２３は、その能力や機能などに応じて予めデバイスドライバに実装することが決められているＤＤＩ２０２２（Device Driver Interface ）と呼ばれる関数群を用意する必要がある。アプリケーションのＡＰＩ（Application Programming Interface）コールについては、ＧＤＩ２０２１がデバイスドライバ用にデータ変換し、上記ＤＤＩ関数群２０２２が適宜ＧＤＩ２０２１からコールされて所定の印刷処理が実行されるようになっている。ＧＤＩ２０２１は、このようにプリンタドライバ２０２３を介してアプリケーションからの印刷要求をシーケンシャルに処理している。一般的には、このようにＯＳの枠組み内にあるプリンタドライバ２０２３内でＰＤＬコマンドを生成したり、イメージモードとしてレンダリング処理を行っている。

【０００７】

また、イメージモード時のバンティング処理を独自で行ったり、逆順印刷等のページハンドリング処理・処理モードの自動切り替えといったＯＳがサポートしていない機能を実現するために、プリンタドライバ２０２３は受け取った描画命

令から中間言語ファイル（ＰＤＦ２０３１）の生成のみを行い、アプリケーションであるスプールサブシステム２０３のデスプーラ２０３２がＰＤＦ２０３１を読み込み、付加機能の追加を行っている。

【０００８】

プリンタドライバの処理は、ＰＤＬモード（２０３３参照）とイメージモード（２０３４～２０３６参照）の２種類に大別できる。

【０００９】

ＰＤＬモード２０３３は、印刷装置側がＰＤＬ（Printer Description Language）と呼ばれる制御コマンドを処理できるコントローラを搭載している場合に用いられ、ＰＤＦ２０３１からＰＤＬのコマンドに変換し次第、インタフェース（Ｉ／Ｆ）部２０３８を介して印刷装置（プリンタ）１００に送出する。

【００１０】

ＰＤＬモード時の処理は、ＤＤＩ２０２２で受け取ったデータをシーケンシャルにコマンド生成するだけであるので、ドライバにとっては軽い処理といえる。ＰＤＬモードタイプでは、プリンタのＰＤＬコマンドの解析能力が高ければＧＤＩ２０２１から受け取った抽象度の高いデータをそのまま送出できるので、文字中心の通常ページ印刷については出力階調に拘わらず高速である、という特徴をもつ。

【００１１】

しかし、グラフィックス系アプリケーションで複雑な図形・高解像度・高階調イメージを出力しようとすると、ＰＤＬコマンドが大きくなり転送速度・プリンタの処理速度が落ちる。理論的にはＰＤＬコマンドサイズに上限はない。

【００１２】

一方、イメージモードは、プリンタドライバ側で確保したバンドメモリ空間上に印刷イメージの展開を行い、それを印刷装置で直接印刷できる形態に変換し、印刷装置に送り印刷するものである。

【００１３】

イメージモードタイプでは、転送データサイズが描画面積と出力階調数で決まるため、簡単なページでも複雑なページでも転送速度・プリンタの処理時間は変

わらない、という特徴がある。そのため、ＰＤＬモードで処理時間のかかるページでも比較的高速な印刷を行うことが可能であるが、文字だけの簡単なページでもプリンタをエンジンスループットで動かすことは難しい。

【００１４】

デスプーラ２０３２は、ＰＤＦ２０３１をデスプールする際に、適当な高さに分割したバンドメモリを用い、複数回に分けて描画位置に関係する描画命令をイメージモード用モジュール２０３４～２０３６に渡して描画処理を行う。生成したビットマップデータは、バンド毎にＩ／Ｆ部２０３８を介して印刷装置１００に送出される。

【００１５】

イメージモードの特徴として、高解像度カラーイメージや複雑なグラフィックスデータの描画処理が高速である点が挙げられる。しかし、文字あるいは簡単な図形についても、描画領域全体のイメージデータを出力解像度・出力階調で生成し、印刷命令として送出するので、高解像度・高階調ページの高速印刷には適していない。

【００１６】

これらの両モードとも、描画処理を行うために、イメージモードではホストコンピュータ側に、ＰＤＬモードではプリンタ側に１ページ分の描画用メモリ領域が必要である。ところが、カラープリンタではＲＧＢ各色８ビットで６００ＤＰＩ，Ａ４フルページ分のメモリを確保しようとすると９６メガバイトが必要であり、現状ではホストマシンでも確保することは困難である。そこで、カラープリンタの描画処理では、出力形式であるＹＭＣＫ各色１，２または４ビットの色空間を使い、さらにページをいくつかの領域に分割したバンドメモリを使うことで、描画処理用メモリの削減を行っている。

【００１７】

このように、１ページを複数のバンドに分割して描画処理を行うことをバンディング処理と呼んでいる。このバンディング処理は、ＰＤＬモードの場合はプリンタ側で、イメージモードの場合はホストマシン側で１ページ分の描画命令を保持し、各バンドに必要な印刷命令のみを処理することで、各バンドの描画処理を

実現している。

【００１８】

【発明が解決しようとする課題】

しかし、上記のＤＤＩ２０２２を介して渡される描画命令の中には、１ライン描画関数(Windows9X;Output 関数のSCANLINE)，数ピクセルの矩形塗りつぶし関数(Windows9X;Output 関数のRECTANGLE)が何度もコールされて、複雑な形の図形描画処理を行う場合がある。こういった場合に、プリンタドライバで１ページ分のスプール処理を行うと、中間言語数・データサイズの増大が起こり、速度低下の要因となってしまうという問題が生じる。

【００１９】

よって本発明の目的は、上述の点に鑑み、プリンタドライバで中間言語を一旦スプールする場合でも、保持データサイズ・個数を抑えることで高速な印刷を可能とした、情報処理装置および情報処理方法を提供することにある。

【００２０】

【課題を解決するための手段】

上記の目的を達成するために、請求項１に係る本発明は、入力された印刷指令情報を格納する記憶手段と、前記記憶手段に前記印刷指令情報を格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記記憶手段に格納させる合成手段とを具備した情報処理装置である。

【００２１】

請求項２に係る本発明は、請求項１に係る情報処理装置において、前記合成手段により、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記記憶手段に格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記記憶記憶手段に格納する。

【００２２】

請求項３に係る本発明は、請求項１に係る情報処理装置において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、

前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【００２３】

請求項４に係る本発明は、請求項３に係る情報処理装置において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【００２４】

請求項５に係る本発明は、請求項１に係る情報処理装置において、前記合成手段は、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【００２５】

請求項６に係る本発明は、請求項１に係る情報処理装置において、前記合成手段は、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求める手段と、前記変化量が前回と同じ場合には繰り返し変数をカウントアップする手段とを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記記憶手段に格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記記憶手段に格納する。

【００２６】

請求項７に係る本発明は、入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを有する情報処理方法である。

【００２７】

請求項８に係る本発明は、請求項７に係る情報処理方法において、前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作成し、その後に、新たな印刷指令情報を前記メモリに格納する。

【００２８】

請求項９に係る本発明は、請求項７に係る情報処理方法において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【００２９】

請求項１０に係る本発明は、請求項９に係る情報処理方法において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【００３０】

請求項１１に係る本発明は、請求項７に係る情報処理方法において、前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【００３１】

請求項１２に係る本発明は、請求項７に係る情報処理方法において、前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップとを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納する。

【００３２】

請求項１３に係る本発明は、入力された印刷指令情報をメモリに格納する際に、既に格納されている印刷指令情報の属性が前記印刷指令情報の属性と同一であって且つ前記記憶手段に格納可能な記憶領域がある場合には、既に格納されている印刷指令情報と前記印刷指令情報を合成して前記メモリに格納させる合成ステップを、読み出し可能なプログラムの形態で記憶した記憶媒体である。

【００３３】

請求項１４に係る本発明は、請求項１３に係る記憶媒体において、前記合成ステップにより、前記印刷指令情報と、既に格納されている印刷指令情報とが合成されない場合には、前記メモリに格納されている印刷指令情報から中間言語を作

成し、その後に、新たな印刷指令情報を前記メモリに格納する。

【００３４】

請求項１５に係る本発明は、請求項１３に係る記憶媒体において、前記印刷指令情報は所定のアプリケーションプログラムを実行させることにより供給され、前記印刷指令情報に基づき１ページ分の中間言語を保持してから、印刷装置に対する印刷命令の生成を行う。

【００３５】

請求項１６に係る本発明は、請求項１５に係る記憶媒体において、前記印刷命令は、所定の通信媒体を介して前記印刷装置に伝送される。

【００３６】

請求項１７に係る本発明は、請求項１３に係る記憶媒体において、前記合成ステップは、同一の属性を持つ印刷指令情報を共通ヘッダにまとめることにより、ヘッダサイズ分の削減を行う。

【００３７】

請求項１８に係る本発明は、請求項１３に係る記憶媒体において、前記合成ステップは、前回保持した描画オブジェクトと現在処理している描画オブジェクトの描画座標変化量を求めるステップと、前記変化量が前回と同じ場合には繰り返し変数をカウントアップするステップとを備え、前記変化量が前回と異なる場合には繰り返し変数を表すカウント数と変化量を示すコマンドを前記メモリに格納し、それに続いて、現在処理している描画オブジェクトの絶対座標を示すコマンドを前記メモリに格納する。

【００３８】

請求項１９に係る本発明は、請求項１３～１８のいずれかに係る記憶媒体において、前記記憶媒体として、サーバコンピュータおよびクライアントコンピュータが読むことができるプログラムを格納したフロッピーディスク、ハードディスク、光磁気ディスク、光ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭを用いる。

【００３９】

請求項２０に係る本発明は、請求項１３～１８のいずれかに係る記憶媒体にお

いて、前記記憶媒体は、サーバコンピュータおよびクライアントコンピュータに着脱可能である。

【００４０】

【発明の実施の形態】

以下、図面を参照して、本発明の実施の形態を詳細に説明する。

【００４１】

図１は、本発明を適用した印刷システムに含まれるカラーレーザービームプリンタ（以下「カラーＬＢＰ」と記す）の断面構成を示す。すなわち本図は、６００ドット／インチ（ｄｐｉ）の解像度を有し、各色成分各画素が８ビットで表現された多値データに基づいて画像記録を行うカラーＬＢＰの構造を示す側断面図である。

【００４２】

図１において、１００はカラーＬＢＰ本体（以下、印刷装置ともいう）であり、外部に接続されているホストコンピュータなどから供給されるプリントデータ（文字コードや画像データ等）および制御コードから成る印刷情報を入力して記憶すると共に、それらの情報に従って対応する文字パターンやイメージ等を作成し、記憶媒体である記録紙上に像を形成する装置である。

【００４３】

１１０はホストコンピュータから供給される印刷情報を解析し印刷イメージの生成処理を行うと共に、カラーＬＢＰ本体１００の制御を行うフォーマッタ制御部である。このフォーマッタ制御部１１０は、ユーザによる操作およびユーザに対する状態通知のためのスイッチおよびＬＥＤ表示器等が配されているオペレーションパネル部１２０と接続されており、そのパネル部は印刷装置１００の外装の一部として配設されている。フォーマッタ制御部１１０において生成された最終的な印刷イメージはビデオ信号として出力制御部１３０に送出される。出力制御部１３０は、印刷装置１００の不図示の各種センサからの状態入力と共に光学ユニット１４０および各種駆動系機構部に対して制御信号を出力し、印刷装置１００としての印刷処理の制御を司る。

【００４４】

図１に示す印刷装置において、給紙カセット１６１から給紙された用紙（転写紙）Ｐはその先端をグリッパ１５４ｆにより挟持されて、転写ドラム１５４の外周に保持される。光学ユニット１４０により感光ドラム１５１上に形成された各色の潜像は、イエロー（Ｙ），マゼンタ（Ｍ），シアン（Ｃ），ブラック（Ｂ）の各色現像器Ｄｙ，Ｄｍ，Ｄｃ，Ｄｂにより現像化されて、転写ドラム外周の用紙に複数回転写され、多色画像が形成される。その後、用紙Ｐは転写ドラム１５４より分離されて、定着ユニット１５５で定着され、排紙部１５９より排紙トレー部１６０に排出される。

【００４５】

各色の現像器Ｄｙ，Ｄｍ，Ｄｃ，Ｄｂは、その両端に回転支軸を有し、各々がその軸を中心に回転可能となるように現像器選択機構部１５２により保持される。これによって、各現像器Ｄｙ，Ｄｃ，Ｄｂ，Ｄｎは、図１に示すように現像器選択のために現像器選択機構部１５２が回転軸１５２ａを中心にして回転しても、その姿勢を一定に維持できる構成をとっている。選択された現像器が現像位置に移動後、支点１５３ｂを中心にして、選択機構保持フレーム１５３がソレノイド１５３ａにより感光ドラム１５１方向へ引っ張られ、感光ドラム１５１方向へ移動して現像処理が行われる。次に、帯電器１５６によって感光ドラム１５１が所定の極性に均一に帯電される。フォーマッタ制御部１１０において画像イメージとして展開された印刷情報は、対応するパターンのビデオ信号に変換され、レーザドライバを介して半導体レーザ１４１を駆動する。入力されたビデオ信号に応じて半導体レーザ１４１から発射されるレーザ光はオンオフ制御され、さらにスキャナモータ１４３によって高速回転するポリゴンミラー１４２で左右方向に振らされ、ポリゴンレンズ１４４、反射鏡１４５を介して感光ドラム１５１上を走査露光する。これにより、感光ドラム１５１上には画像パターンの静電潜像が形成されることになる。次に、例えば、Ｍ（マゼンタ）色の静電潜像がＭ（マゼンタ）色の現像器Ｄｍにより現像され、感光体ドラム１５１上にＭ（マゼンタ）色の第１のトナー像が形成される。

【００４６】

一方、所定のタイミングで転写紙Ｐが給紙され、トナーと反対極性（例えばプ

ラス極性）の転写バイアス電圧が転写ドラム１５４に印加され、感光体ドラム１５１上の第１トナー像が転写紙Ｐに転写されると共に、転写紙Ｐが転写ドラム１５４の表面に静電吸着される。その後、感光ドラム１５１はクリーナー１５７によって残留するＭ（マゼンタ）色トナーが除去され、次の色の潜像形成および現像行程に備える。

【００４７】

以下同様の手順によってＣ（シアン），Ｙ（イエロー），Ｂｋ（ブラック）の順で第２，３，４色目のトナー像の転写が行われる。但し、各色の転写時には、転写ドラム１５４には前回よりも高いバイアス電圧が印加される点が異なる。４色のトナー像が重畳転写された転写紙Ｐの先端部が分離位置に近づくと、分離爪１５８が接近してその先端が転写ドラム１５４の表面に接触し、転写紙Ｐを転写ドラム１５４から分離させる。分離された転写紙Ｐは定着ユニット１５５に搬送され、ここで転写紙上のトナー像が定着されて排紙トレイ１６０上に排出される。

【００４８】

本実施の形態におけるカラーレーザビームプリンタは、以上のような画像形成過程を経て６００ドット／インチ（ｄｐｉ）の解像度で画像出力を行う。なお、本発明を適用可能なプリンタは、カラーＬＢＰに限られるものではなく、インクジェットプリンタやサーマルプリンタ等、他のプリント方式のカラープリンタでもよい。

【００４９】

次に、カラーＬＢＰ（印刷装置）１００におけるフォーマッタ制御部１１０について図３を用いて説明する。このフォーマッタ制御部１１０は、通常はＰＤＬコントローラなどとも呼ばれており、ホストコンピュータとの接続手段であるインターフェース（Ｉ／Ｆ）部１１１と、受信データ等を一時的に保持管理するための受信バッファ１１２１と、送信データ等を一時的に保持管理するための送信バッファ１１２２と、印刷データの解析を司るコマンド解析部１１３と、印刷制御処理実行部１１４と、描画処理実行部１１５と、ページメモリ１１６とを含んでいる。

【００５０】

インターフェース（Ｉ／Ｆ）部１１１は、ホストコンピュータ２００との印刷データの送受信を行う通信手段あり、通信プロトコルとしてＩＥＥＥ－１２８４に準拠した通信を可能とするものである。但し、本実施の形態では、この通信手段に限定するものでなく、ネットワークを介してさまざまなプロトコルによる接続であってもよいし、ＩＥＥＥ－１３９４に準じた通信手段であってもよい。このインターフェース部１１１を通して受信した印刷データは、そのデータを一時的に保持する記憶手段である受信バッファ１１２１に逐次蓄積され、必要に応じてコマンド解析部１１３または描画処理実行部１１５によって読み出され処理される。

【００５１】

コマンド解析部１１３は、各ＰＤＬコマンド体系や印刷ジョブ制御言語に準じた制御プログラムにより構成されており、文字印字、図形、イメージなどの描画に関する印刷データの解析結果は、描画処理実行部１１５に指示を与えて処理し、給紙選択やリセット命令などの描画以外のコマンドは、印刷制御処理実行部１１４に指示を出し処理する。

【００５２】

描画処理実行部１１５は、文字やイメージの各描画オブジェクトをページメモリ１１６に逐次展開して行くレンダラである。

【００５３】

図１に示したカラーＬＢＰに対しては、ＭＣＹＫの面順次でエンジン回転処理に間に合うようにページデータを送出する必要があるが、標準状態では、そのために必要なメモリをすべて確保するわけではなく、１プレーン（１または２bit/pixel）の数分の１のバンド領域としてメモリが確保され、そのバンド領域を使いまわして画像をエンジン速度に同期して処理するように構成してある。通常は、このようにＹＭＣＫレンダラによる展開処理とプリンタエンジンへのビデオ信号のシッピングとの追いかけ、所謂バンディング制御によってページメモリ１１６は管理されているが、十分なメモリがある場合は、１ページ分が展開可能な領域を確保しても良い。

【００５４】

なお、一般的に、フォーマッタ制御部１１０は、中央演算処理装置（ＣＰＵ）、リードオンリーメモリ（ＲＯＭ）、ランダムアクセスメモリ（ＲＡＭ）などを用いたコンピュータシステムによって構成されている。また、各部の処理は、マルチタスクモニタ（リアルタイムＯＳ）のもとでタイムシェアリングに処理される構成であっても良いし、各機能ごとに専用のコントローラハードウェアを用意して独立して処理される構成であってもかまわない。

【００５５】

オペレーションパネル１２０は、前述した通り印刷装置の各種状態を設定・表示するためのものである。出力制御部１３０は、ページメモリ１１６の内容をビデオ信号に変換処理し、プリンタエンジン部１５０への画像転送を行う。プリンタエンジン部１５０は受け取ったビデオ信号を記録紙に永久可視画像形成するための印刷機構部であり、図１において前述したものである。

【００５６】

以上、ＬＢＰプリンタ（印刷装置）１００について説明したが、次にホストコンピュータ３００を含む印刷システムの全体構成について説明していく。

【００５７】

図３において、３００はホストコンピュータであり、プリントデータおよび制御コードから成る印刷情報を印刷装置１００に出力する。ホストコンピュータ３００は、入力デバイスであるキーボード３１０や、ポインティングデバイスであるマウス３１１と、表示デバイスであるディスプレイモニタ３２０を合わせた一つのコンピュータシステムとして構成されている。このホストコンピュータ３００は、Windows ＮＴ、Windows95(98) などの基本ＯＳによって動作しているものとする。

【００５８】

ホストコンピュータ側について、本実施の形態に関する機能的な部分にのみ注目し、基本ＯＳ上での機能を大きく分類すると、アプリケーションソフトウェア３０１、グラフィックサブシステム３０２、印刷情報格納手段および印刷装置との通信手段を含むスプールサブシステム３０３に大別される。アプリケーション

ソフトウェア３０１は、例えば、ワープロや表計算などの基本ソフトウェア上で動作する応用ソフトウェアを指すものである。グラフィックサブシステム３０２は、基本ＯＳの機能の一部であるGraphic Device Interface（以後、ＧＤＩと記す）３０２１とそのＧＤＩから動的にリンクされるデバイスドライバであるところのプリンタドライバ３０２２によって構成されている。

【００５９】

プリンタドライバ３０２３は、ＧＤＩからＤＤＩ（Device Driver Interface）３０２２というインターフェースを介してコールされ、デバイスに応じた処理を描画オブジェクト毎に行うものである。本システムでは、プリンタドライバが、ＤＤＩ関数を介して受け取った描画命令の情報を、描画オブジェクトタイプ毎のタグをつけて、描画位置情報、色情報、パターン情報、描画論理情報といった属性と共に、一次バッファ３０２４に保持する。

【００６０】

但し、一次バッファ３０２４に入りきらないような描画オブジェクト（巨大なイメージデータ等）は保持しない。

【００６１】

一次バッファ３０２４に保持する際に、既に保持されている描画オブジェクトがある場合、タイプ、描画属性が同じ時は一次バッファ３０２４の中に追加登録していく。「タイプが異なる。」「一次バッファにデータが入りきらない」、「描画属性が変化する」、「ページ処理終了」の何れかに当てはまるときは、既に一次バッファ３０２４に入っている描画オブジェクトからＰＤＦ３０３１（ページデータファイル）を生成し、スプールする。

【００６２】

スプールサブシステム３０３は、グラフィックサブシステム３０２の後段に位置するプリンタデバイスに特有のサブシステムであり、デスプーラ３０３２を含んでいる。デスプーラ３０３２は、印刷情報格納手段であるところのＰＤＦ３０３１を読み込み、指定されたモード用トランスレータ（３０３３～３０３６）でデスプール処理を行うものである。選択されたドライバ（３０３３～３０３６）は、イメージモード（３０３４～３０３６）であれば、バンドメモリ３０３７を

用いてイメージ展開を行い、ＰＤＬモード（３０３３）であればスプールファイルからＰＤＬコマンドへの変換処理を行い、Ｉ／Ｆ部３０３８を介してプリンタにデータを送出する。

【００６３】

基本ＯＳによって、上述したこれらの名称や機能的な枠組みは若干異なる場合があるが、本実施の形態で言う各技術的手段が実現できるモジュールであれば、それらの名称や枠組みはあまり大きな問題ではない。例えば、スプーラやスプールファイルと呼ばれるものは、別のＯＳにおいてプリントキューと呼ばれるモジュールに処理を組み込むことによっても実現可能である。

【００６４】

なお一般的に、これらの各機能モジュールを含むホストコンピュータ３００は、中央演算処理装置（ＣＰＵ）、リードオンリーメモリ（ＲＯＭ）、ランダムアクセスメモリ（ＲＡＭ）、ハードディスクドライブ（ＨＤＤ）、各種入出力制御部（Ｉ／Ｏ）などのハードウェアのもとで、基本ソフトと呼ばれるソフトウェアがその制御を司り、その基本ソフトの元で、それぞれの応用ソフト、サブシステムプロセスが機能モジュールとして動作するようになっている。

【００６５】

次に、一次バッファを使ったＰＤＦデータ圧縮方法の更に詳しい説明を行う。

【００６６】

ここでは、本実施の形態における動作を説明するために、Windows95(Windows98)のＤＤＩの説明をしておく。

【００６７】

Windows9XのＤＤＩにはグラフィックス系関数の一つとして、

```
Output(LPPDEVICE lpDestDev, WORD wStyle,
       WORD wCount,   LPPOINT lpPoints,
       LPPEN lpPPen,  LPPBRUSH lpPBrush,
       LPDRAWMODE     lpDrawMode,
       LPRECT   lpClipRect )
```

が用意されている。

【００６８】

本関数は、第二引数のwStyleの値によって、渡されるデータの意味が変化する。例えば、OS_RECTANGLEがセットされていた場合、wCountには必ず２がセットされる。本関数は第４引数のlpPointsが示す最初の座標情報（Ｘ，Ｙ）は矩形の左上を、次の座標情報（Ｘ，Ｙ）は右下の点を示し、その対角線を持つ矩形の内側をlpPBrushが示す色、パターン、lpPPenが示す枠線で、描画処理することが求められる。

【００６９】

プリンタドライバ３０２３により、ＤＤＩ関数毎にＰＤＦを生成すると図１０に示す１０００のようなデータ構造のファイルが生成される。本図において、１０００はヘッダ部１００１とデータ部１００２で構成されている。ヘッダ部１００１の先頭にはデータタイプ１００３を持っている。ヘッダ部１００１にはその他にブラシ情報１００４、ペン情報１００５、データサイズCount,論理演算値１００６、外接矩形情報１００７等を持っている。データ部１００２は、ＰＯＩＮＴ型構造体配列で構成されており、Count １００５数分の配列にデータが格納されている。

【００７０】

しかし、グラフィックス系アプリケーションでグラデーションを描いて出力しようとすると、図４の４００で示すような細かなRECTANGLE を多量に指定してくる場合がある。すなわち、４００のＡ行-[５～６] 列が４０１、Ｂ行-[４～６] 列が４０２、Ｃ行-[３～６] 列が４０３・・・である。

【００７１】

このように、ＤＤＩ関数に渡された細かな単位で矩形描画命令を保持するとＰＤＦのサイズが膨れ上がり、ＰＤＬモード、イメージモード共に印字速度の低下に繋がる。そこで、本実施の形態は、プリンタドライバ３０２３ではＤＤＩ関数がコールされても直ちにＰＤＦを生成せずに、固定サイズのメモリ空間である一次バッファ３０２４を使うことでＰＤＦの圧縮処理を行う。

【００７２】

一次バッファ３０２４は、ページ内で保持される必要があり、かつドライバが

提供している全てのＤＤＩ関数からもアクセス可能でなくてはならない。Windows9Xの全てのＤＤＩには“LPPDEVICE lpDestDev”が第一引数に入っており、一つのジョブ内で保持されるメモリ空間へのポインタを示している。一次バッファへのポインタはメンバとして登録する。

【００７３】

本実施の形態における圧縮処理は、大きく分けて以下の二つの手法の組み合わせによって実現する。

【００７４】

１）　ヘッダ部の共通化によるサイズ、コマンド数の削減

２）　データ部圧縮によるデータサイズの削減

以下、それぞれについての詳しい説明を行う。

【００７５】

まず、「１）　ヘッダ部の共通化によるサイズ、コマンド数の削減」について説明する。

【００７６】

細かく分割された同じ属性の描画命令は、ヘッダ情報の共通化でＰＤＦのサイズ削減が可能である。

【００７７】

前回の描画属性と比較するために、以下のようなメンバーが必要である。

【００７８】

```
typedef struct
{
    enum ELEMTYPE lasDataType;    //前回データタイプ
    WORD lastPenIndex;            //前回データのペン情報
    WORD lastBrushIndex;          //前回データのブラシ情報
    DRAWMODE  lastDrawMode;       //前回データのDrawMode
    RECT    lastboundRect;        //前回データの外接矩形
    RECT    BoundRect;            //圧縮データ全体の外接矩形
    LPSTR  lpstart;               //圧縮データの先頭ポインタ
```

```
    LPSTR   lplast;                 //圧縮データの現状の最終ポインタ
    WORD    bufsize;                //一次バッファサイズ
} LASTDATA;
```

【００７９】

プリンタドライバが提供するＤＤＩ関数内では、上記LASTDATAにセットされている前回のデータタイプと今回のデータを比較し、lasDataType が同じときは、タイプ別圧縮処理関数をコールし、違う場合は一次バッファ内のデータからＰＤＦの生成を行う。

【００８０】

図５は、以上の処理の流れを示すフローチャートである。図５における各ステップの処理内容は、以下の通りである。

【００８１】

ステップＳ６０１：一次バッファ内に描画データが格納されているならステップＳ６０６へ、一次バッファにデータがセットされていない場合はステップＳ６０２へ制御を移す。

【００８２】

ステップＳ６０２：一次バッファ内に格納した描画データのタイプ、属性（ブラシ、ペン、論理演算種）と同じならばステップＳ６０３へ、違う場合は６０４へ制御を移す。

【００８３】

ステップＳ６０３：一次バッファ内に渡されたデータを格納可能ならばステップＳ６０５へ、不可能ならステップＳ６０４へ制御を移す。

【００８４】

ステップＳ６０４：一次バッファに格納されている描画データからＰＤＦを作製し、ステップＳ６０５へ制御を移す。

【００８５】

ステップＳ６０５：現在のＤＤＩ関数に渡された描画データをデータタイプ別圧縮関数に渡して一次バッファにセットする。その後、ステップＳ６０６へ制御を移す。

【００８６】

ステップＳ６０６：LASTDATAを更新し、ステップＳ６０７へ制御を移す。

lastDataType	変更があった場合はセット
lastPenIndex	変更があった場合はセット
lastBrushIndex	変更があった場合はセット
lastDrawMode	変更があった場合はセット
lastboundRect	今回の描画命令の外接矩形をセット
BoundRect	前回のBoundRect と今回の外接矩形のAND を取る。
lpstart	一次バッファの先頭アドレス、変更なし
lplast	一次バッファ内の現在のデータを詰めこんだ最終ポインタ
bufsize	今回詰めこんだデータサイズを足しこむ。

【００８７】

ステップＳ６０７：描画命令全て終了ならステップＳ６０８へ、まだならステップＳ６０１へ制御を移す。

【００８８】

ステップＳ６０８：一次バッファ内に残っている描画データからＰＤＦを作製する。

【００８９】

図６は、上記のステップＳ６０４またはステップＳ６０８によって得られた結果を例示したものである。

【００９０】

次に、「２）　データ部圧縮によるデータサイズの削減」について説明する。

【００９１】

一次バッファ３０２４に既に格納されている描画命令と同一タイプの描画命令がコールされた時にコールされるデータタイプ別圧縮関数は、座標空間の連続性等を活かしてデータ部の圧縮処理を行う。

【００９２】

具体例として、Windows9X，Output(Scanline)の圧縮方法の一例を示す。

【００９３】

１．Output(Scanline)の圧縮アルゴリズムについて

図７を参照しながらOutput(Scanline)関数の説明を行う。ＤＤＩ関数の一つである　Output関数の第２引数にOS_SCANLINESがセットされているときには、lpPointsが示す先のPOINT 構造体配列は、

lpPoints [0]->y ： Ｙ座標

lpPoints [1]->x ： 左座標、　lpPoints [1]->y ： 右座標

を示す。

wCount は lpPoints配列の個数を示すが、スキャンラインが１本の場合は２，２本の場合は３，ｎ本の場合はｎ＋１がセットされている。

【００９４】

図８は、Output関数で複数回Scanline描画指定を受けた時に、圧縮処理を行わない場合と、以上二つの手法を組み合わせた場合の動きを示す模式図である。描画領域８００に図形８０２を描画する場合、Windows9XはOutput関数８０１を使ってＹ座標の小さい方から順にScanline描画を指定してくる場合がある。

【００９５】

一つのOutput関数に対して一つのヘッダを持つＰＤＦ８０３を作ると、同一のブラシやペンといった重複データが何度も登録されることになる。

【００９６】

図８の８０４は、第一の方法によって、同一ブラシ、ペン、描画属性はこれらの情報を一つのヘッダとして扱うことでヘッダサイズの削減が可能であることを示している。

【００９７】

さらに、図８の８０２のようにScanline同士が接している場合、以下の方法によって全座標情報を保持するよりも、データサイズを小さくすることが可能である。

【００９８】

まず、Scanlineデータ圧縮時には一次バッファ管理用にLASTDATA以外にも以下のメンバーを保持する。

【００９９】

```
typedef Struct
{
    WORD repetcnt;          // 繰り返し回数
    WORD lastY;             // 前回のＹ座標
    WORD lastLeftX;         // 前回の左Ｘ座標
    WORD lastRightX;        // 前回の右Ｘ座標
    WORD delLeftX;          // 前回の左Ｘ座標変化量
    WORD delRightX;         // 前回の右Ｘ座標変化量
} CMPSCNLINES;
```

CMPSCNLINES にそれまでのScanlineの情報を蓄えることにより、図１１～図１４に示すコマンドを生成する。

【０１００】

次に、コマンド使用例について説明する。

【０１０１】

圧縮コマンドは基本的に、［初期座標］［変化分コマンド］の連続である。

【０１０２】

以下に、コマンド排出例を示す。

【０１０３】

１．縦一直線

縦一直線の場合は、「長さ分連続する変化分０のSCANLINE」となる。

従ってコマンドは、

［（番号０）（初期座標）］

［（番号１５）（連続数＝長さ）］

［（番号３）（左変化分＝０）（右変化分＝０）］

となる。

【０１０４】

２．斜め４５度線

斜め４５度線の場合は、左右変化分が１となる場合がほとんどである。

よって「長さ分連続する変化分１のSCANLINE」となるので、コマンドは、

［（番号０）（初期座標）］

［（番号１５）（連続数＝長さ）］

［（番号３）（左変化分＝１）（右変化分＝１）］

となる。

【０１０５】

３．斜め線

一般的な斜め線では、各SCANLINE毎の変化はまちまちである。

そこで変化分が変わったその都度コマンドを発行する。

［（番号０）（初期座標）］

［（番号３）（左変化分）（右変化分）］

［（番号３）（左変化分）（右変化分）］

［（番号３）（左変化分）（右変化分）］

：

となる。

【０１０６】

図９は、以上のコマンド生成処理の流れを示すフローチャートである。図９における各ステップの処理内容は、以下の通りである。

【０１０７】

ステップＳ９０１：一次キャッシュ内に格納されている前回のScanlineデータのＹ座標LastYと、今回のScanlineのＹ座標の差delYを求める（一次キャッシュ内に未格納時はdelY＝０）。

【０１０８】

ステップＳ９０２：delYが０以外ならステップＳ９０５へ、０ならステップＳ９０３へ制御を移す。

【０１０９】

ステップＳ９０３：今回のScanlineの左ＸとlastLeftX，右Ｘと lastRightXとの差分を求める。

【０１１０】

ステップＳ９０４：左右とも前回の差分 delLeftX，delRightXと同じならス

テップＳ９０９へ、どちらか一つでも違う場合はステップＳ９０５へ制御を移す。

【０１１１】

ステップＳ９０５：繰り返しカウンタrepetcnt が０ならステップＳ９０６へ、それ以外ならばステップＳ９０７へ制御を移す。

【０１１２】

ステップＳ９０６：連続コマンド(CmndNo.15) を一次バッファ３０２４にバッファリングし、続けて delLeftX, delRightXを示すコマンドをバッファリングし、ステップＳ９０７へ制御を移す。

【０１１３】

ステップＳ９０７：新規座標コマンド(CmndNo.０）でＹ座標、左右Ｘ座標を一次バッファ３０２４にバッファリングし、ステップＳ９０８へ制御を移す。

【０１１４】

ステップＳ９０８：lastLeftX,lastRightXにステップＳ９０７でセットした左右Ｘ座標をセットし、 delLeftX, delRightXは０で初期化する。

【０１１５】

ステップＳ９０９：繰り返しカウンタrepetcnt をカウントアップし、ステップＳ９１０へ制御を移す。

【０１１６】

ステップＳ９１０：lastLeftX,LastRightXを更新する。

【０１１７】

なお、本発明は、複数の機器（例えばホストコンピュータ、インターフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、１つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用しても良い。

【０１１８】

また、本発明の目的は、前述した実施形態の機能を実現するプリンタドライバと呼ばれるソフトウェアプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読み出し実行することによ

っても、達成されることは言うまでもない。

【０１１９】

この場合、記憶媒体から読み出されたプログラムコード事態が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【０１２０】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ，ＣＤ－Ｒ，磁気テープ、不揮発性のメモリカード、ＲＯＭなどを用いることができる。

【０１２１】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１２２】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【０１２３】

本発明を記憶媒体に適応する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになる。

【０１２４】

本発明は、前述した実施の形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体からそのプログラムをパソコン通信など通信ラインを介して要求者にそのプログラムを配信する場合にも適用できることは言うまでもな

い。

【０１２５】

【発明の効果】

以上説明したように、本発明によれば、プリンタドライバで中間言語を一旦スプールする場合、保持データサイズ・個数を抑えることで高速な印刷が可能になる。すなわち、同一図形をＯＳが別関数で描画指定をしてきた場合も、スプールオブジェクト個数、スプールデータサイズを抑えることにより、高速な印刷処理が可能となる。

【図面の簡単な説明】

【図１】

本発明を適用したレーザビームカラープリンタ（ＬＢＰ）の構造を示す側断面図である。

【図２】

従来の描画方法を示す概念図である。

【図３】

本発明を適用した印刷システムの全体的基本構成を示すブロック図である。

【図４】

RECTANBLE 描画命令を個々にヘッダを付けてＰＤＦに落とした様子を示す模式図である。

【図５】

描画属性が同じ描画命令のヘッダを共通化する処理系を示すフローチャートである。

【図６】

本実施の形態によって複数に分けて渡されたScanline描画命令に一つのヘッダを付けてＰＤＦに落とした様子を示す模式図である。

【図７】

Windows9Ｘ：ＤＤＩの一つであるOutput関数(Scanline)と、パラメータの関係を示す模式図である。

【図８】

Windows9XのScanlineで図形を描画した様子と、従来のＰＤＦおよび本実施の形態によるＰＤＦの違いを示す模式図である。

【図９】

別関数で送られてきたScanlineデータを圧縮する手順を示すフローチャートである。

【図１０】

従来のScanline描画命令のＰＤＦ構造体を示す図である。

【図１１】

本実施の形態で用いるコマンドの説明図である。

【図１２】

本実施の形態で用いるコマンドの説明図である。

【図１３】

本実施の形態で用いるコマンドの説明図である。

【図１４】

本実施の形態で用いるコマンドの説明図である。

【符号の説明】

１００　印刷装置（ＬＢＰ）
１１０　フォーマッタ制御部
１１１　インターフェース部
１１２１　受信バッファ
１１２２　送信バッファ
１１３　コマンド解析部
１１４　印刷制御処理実行部
１１５　描画処理実行部（レンダラ）
１１６　ページメモリ
１２０　オペレーションパネル
１３０　出力制御部
１４０　光学ユニット
１４１　半導体レーザ

１４２　ポリゴンミラー
２００　ホストコンピュータ
２０１　アプリケーションソフト
２０２　グラフィックサブシステム
２０２１　ＧＤＩ
２０２２　ＤＤＩ　Ｉ／Ｆ
２０２３　プリンタドライバ
２１０　キーボード
２１１　マウス
２２０　ディスプレイモニタ
２０３　スプールサブシステム
２０３１　ＰＤＦ
２０３２　デスプーラ
２０３３　ＰＤＬトランスレータ
２０３４　イメージレンダラー（１ＢＰＰ）
２０３５　イメージレンダラー（ＹＭＣＫ４ＢＰＰ）
２０３６　イメージレンダラー（ＲＧＢ２４ＢＰＰ）
２０３７　バンドメモリ
２０３８　Ｉ／Ｆ部
３００　ホストコンピュータ
３０１　アプリケーションソフト
３０２　グラフィックサブシステム
３０２１　ＧＤＩ
３０２２　ＤＤＩ　Ｉ／Ｆ
３０２３　プリンタドライバ
３０２４　一次バッファ
３０３　スプールサブシステム
３０３１　ＰＤＦ（プリントデータファイル）
３０３２　デスプーラ

３０３３　　ＰＤＬトランスレータ

３０３４　モノクロ１ＢＰＰイメージレンダラー

３０３５　　ＹＭＣＫ４ＢＰＰイメージレンダラー

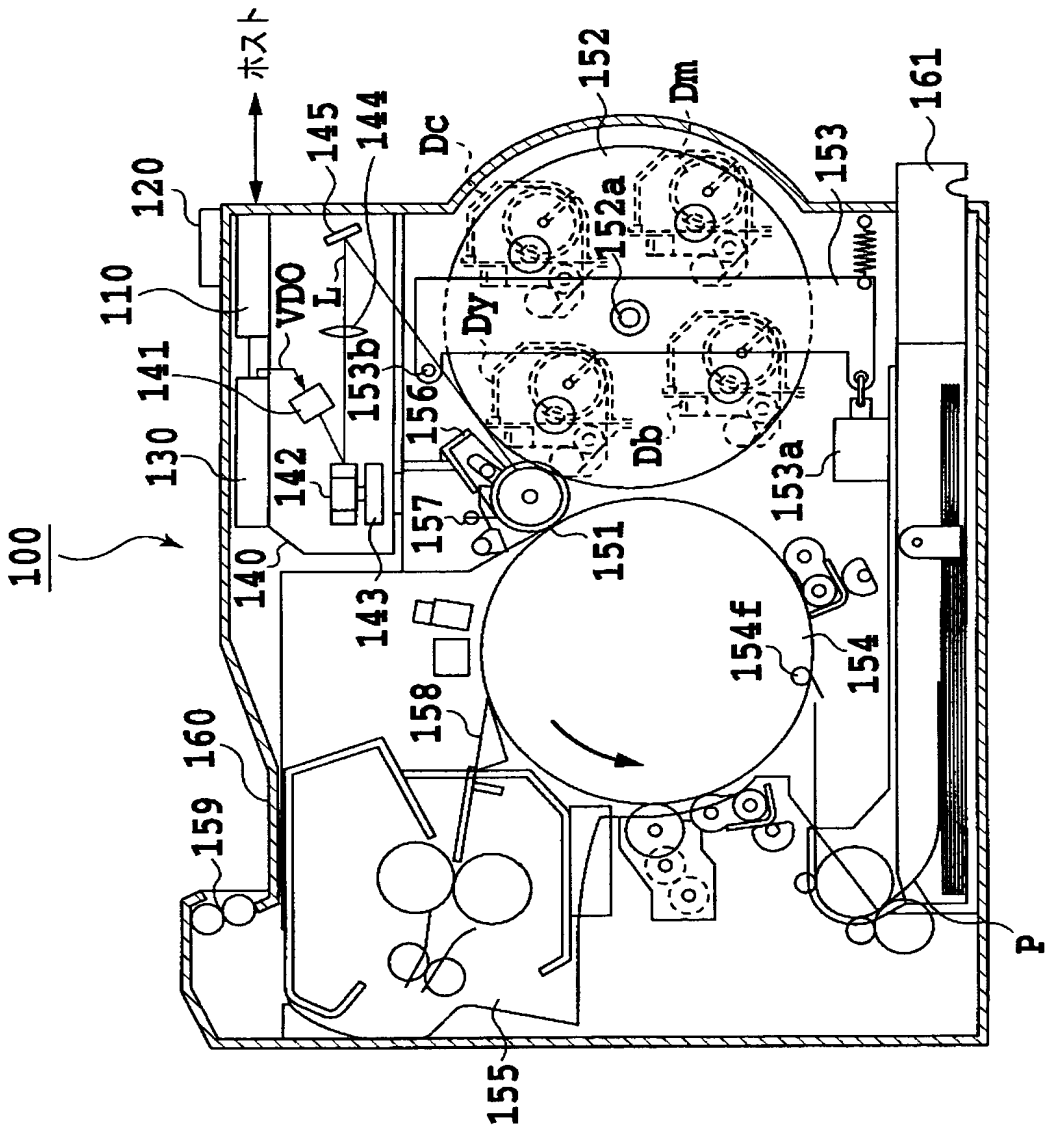
３０３６　　ＲＧＢ２４ＢＰＰイメージレンダラー

３０３７　　イメージモード用バンドメモリ領域

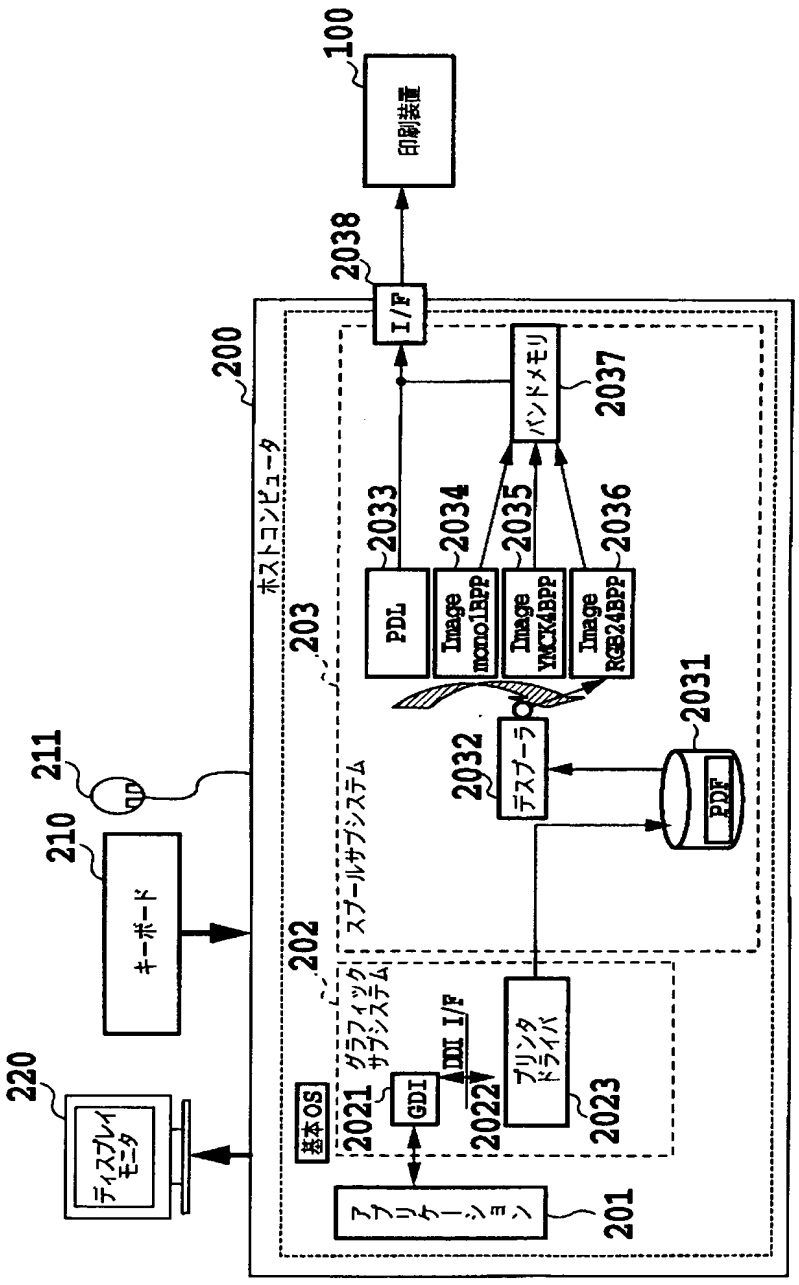
３０３８　　インターフェース部

【書類名】　　図面

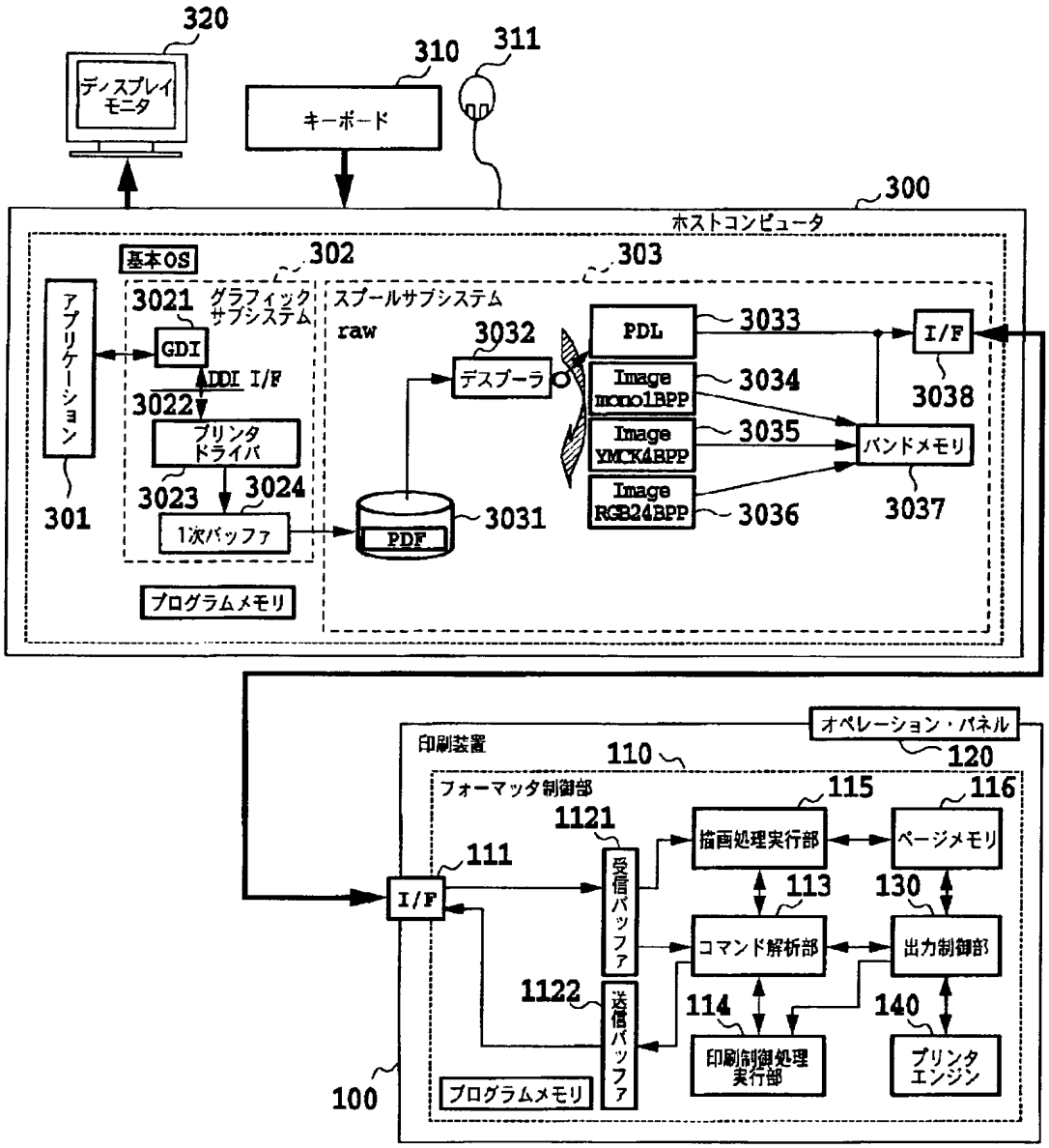
【図１】


ホスト
120
110
141
130
100
140
142
143
145
144
VDO
L
153b
156
157
158
159
160
155
Dc
Dy
Db
Dm
152
152a
151
153
153a
154
154f
161
P

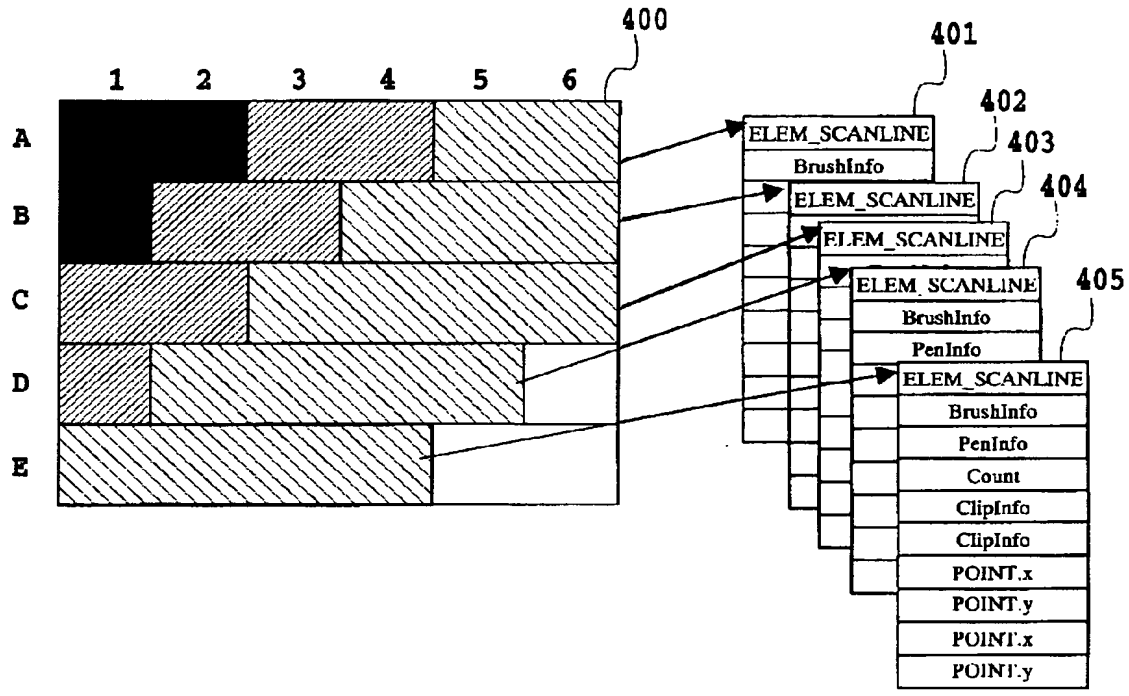
【図２】


100
印刷装置
2038
I/F
200
ホストコンピュータ
2037
バンドメモリ
2033
2034
2035
2036
PDL
Image mono1BPP
Image YMCK4BPP
Image RGB24BPP
203
スプールサブシステム
2031
2032
デスプーラ
PDF
211
210
キーボード
202
グラフィックサブシステム
基本OS
2021
GDI
DDI I/F
2022
プリンタドライバ
2023
220
ディスプレイモニタ
アプリケーション
201

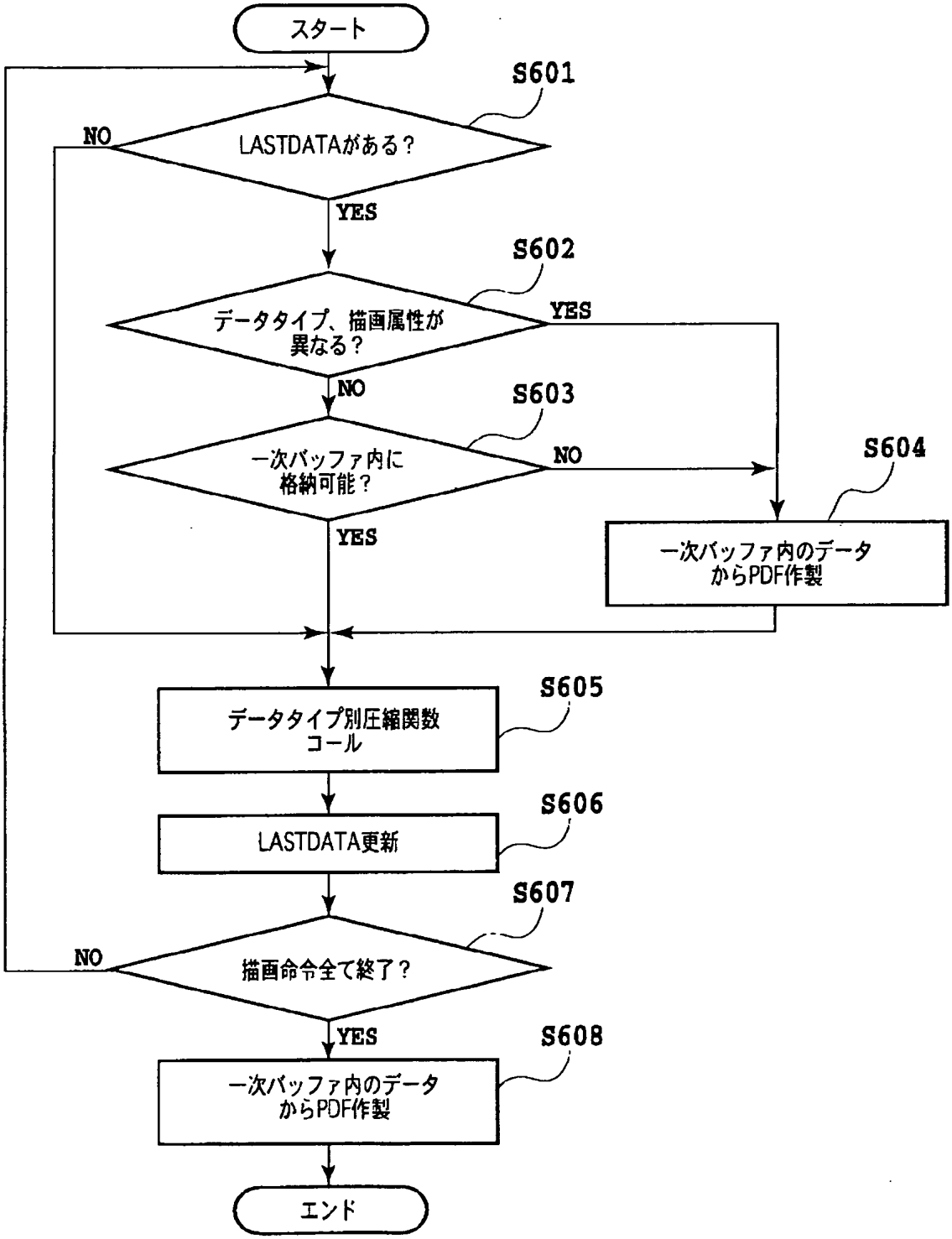
【図３】


320
ディスプレイモニタ
310
キーボード
311
300
ホストコンピュータ
基本OS
302
303
アプリケーション
3021
グラフィックサブシステム
GDI
DDI I/F
3022
プリンタドライバ
3023
3024
1次バッファ
301
スプールサブシステム
raw
3032
デスプーラ
PDF
3031
PDL
3033
Image mono1BPP
3034
Image YMCK4BPP
3035
Image RGB24BPP
3036
I/F
3038
バンドメモリ
3037
プログラムメモリ
オペレーション・パネル
120
印刷装置
110
フォーマッタ制御部
115
116
1121
描画処理実行部
ページメモリ
111
I/F
受信バッファ
113
130
コマンド解析部
出力制御部
1122
送信バッファ
114
140
印刷制御処理実行部
プリンタエンジン
100
プログラムメモリ

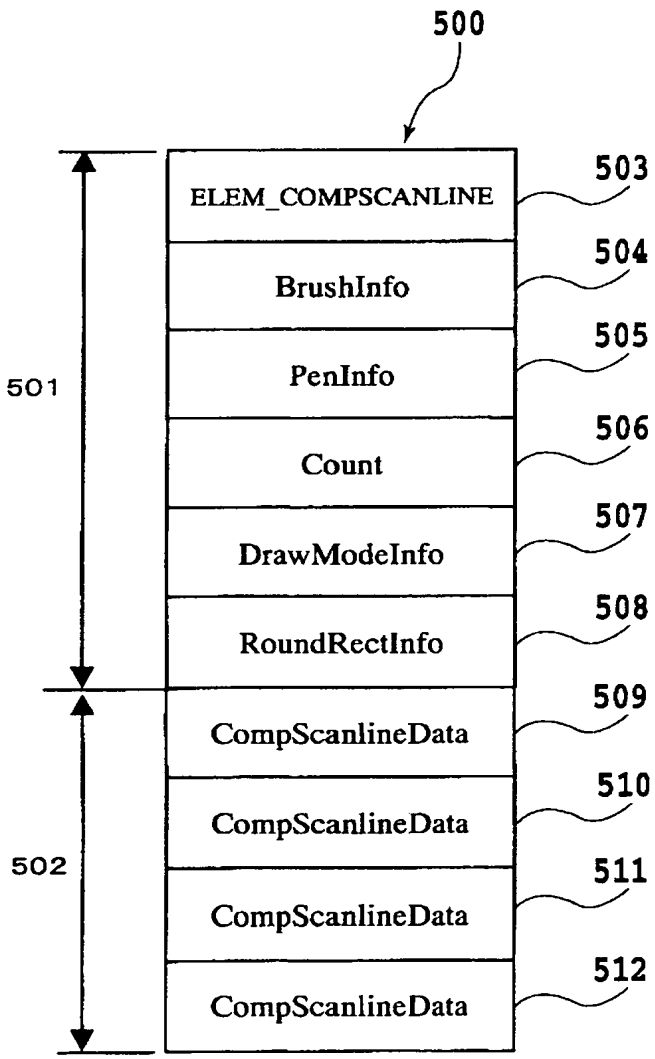
【図４】


400
401
402
403
404
405
1
2
3
4
5
6
A
B
C
D
E
ELEM_SCANLINE
BrushInfo
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
BrushInfo
PenInfo
ELEM_SCANLINE
BrushInfo
PenInfo
Count
ClipInfo
ClipInfo
POINT.x
POINT.y
POINT.x
POINT.y

【図５】


スタート
S601
LASTDATAがある？
NO
YES
S602
データタイプ、描画属性が異なる？
YES
NO
S603
一次バッファ内に格納可能？
NO
YES
S604
一次バッファ内のデータからPDF作製
S605
データタイプ別圧縮関数コール
S606
LASTDATA更新
S607
描画命令全て終了？
NO
YES
S608
一次バッファ内のデータからPDF作製
エンド

【図６】


500
ELEM_COMPSCANLINE
503
BrushInfo
504
PenInfo
505
Count
506
DrawModeInfo
507
RoundRectInfo
508
CompScanlineData
509
CompScanlineData
510
CompScanlineData
511
CompScanlineData
512
501
502

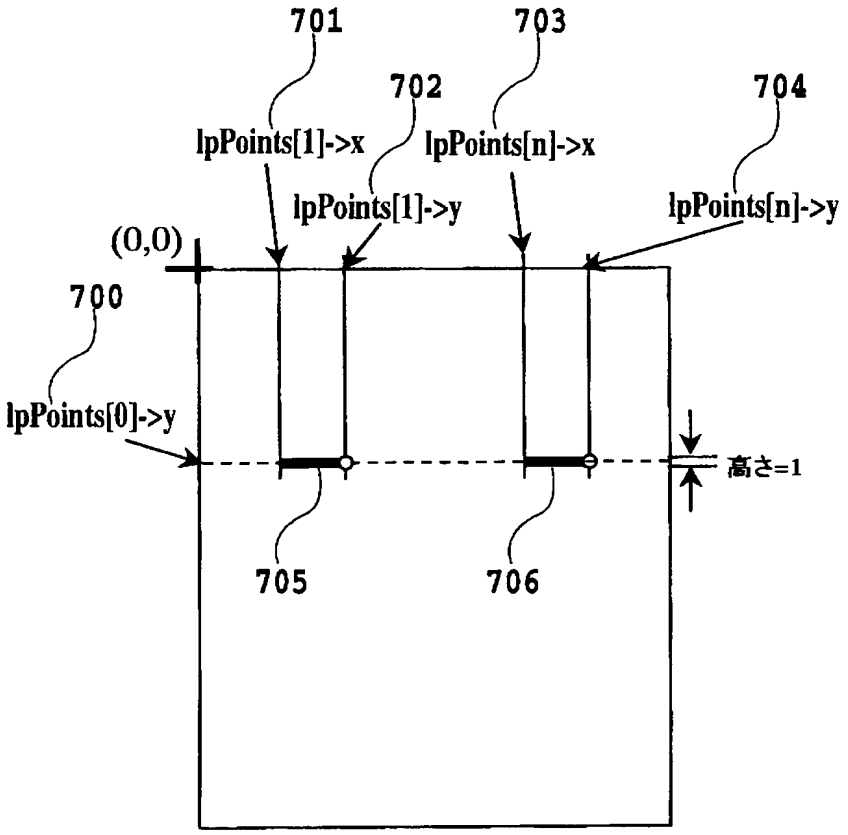
【図７】

関数名:

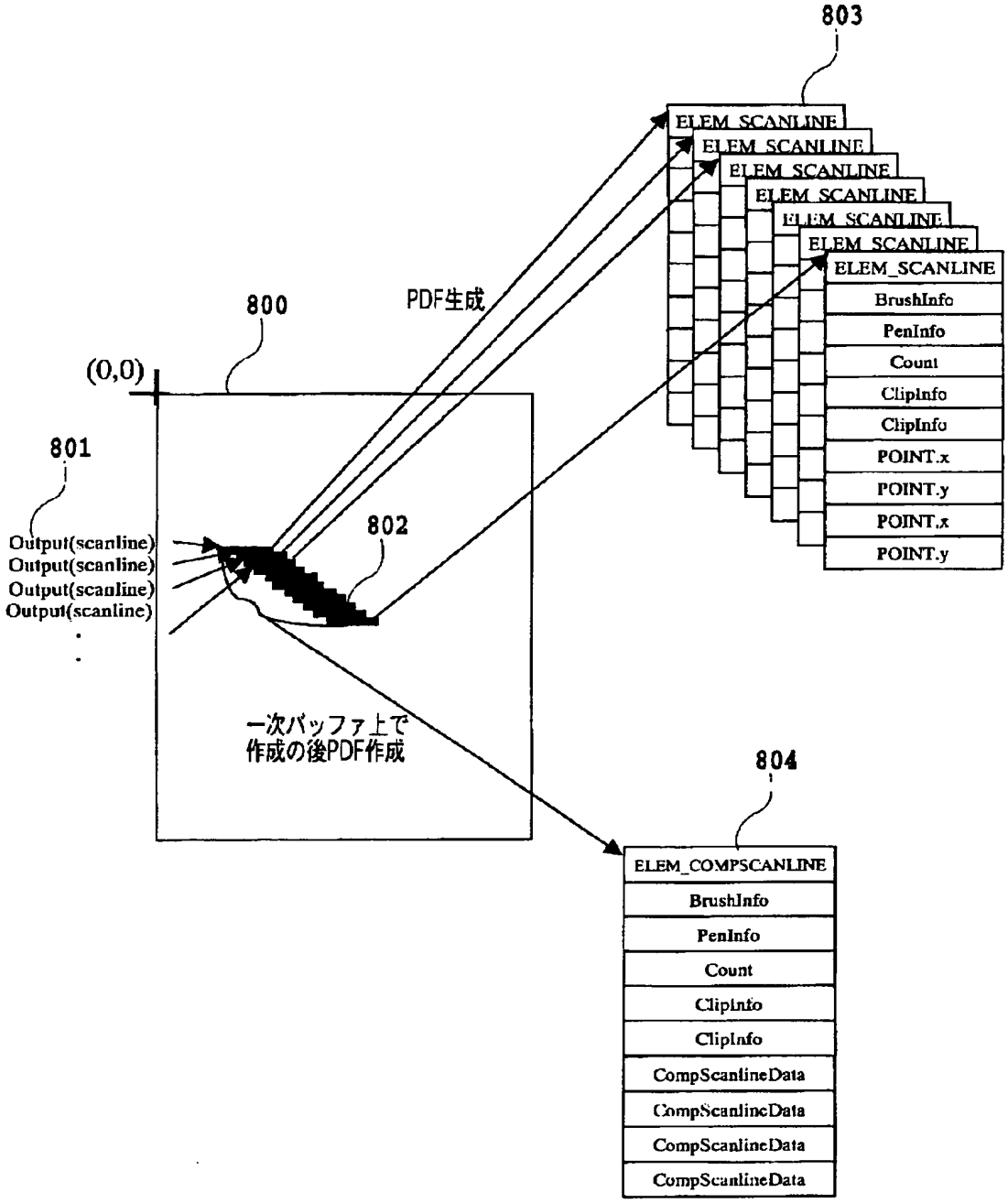
Output(LPPDEVICE lpDestDev、WORD wStyle,
WORD wCount, LPPOINT lpPoints,
LPPEN lpPPen, LPPBRUSH lpPBrush,
LPDRAWMODE lpDrawMode,
LPRECT lpClipRect)

引数

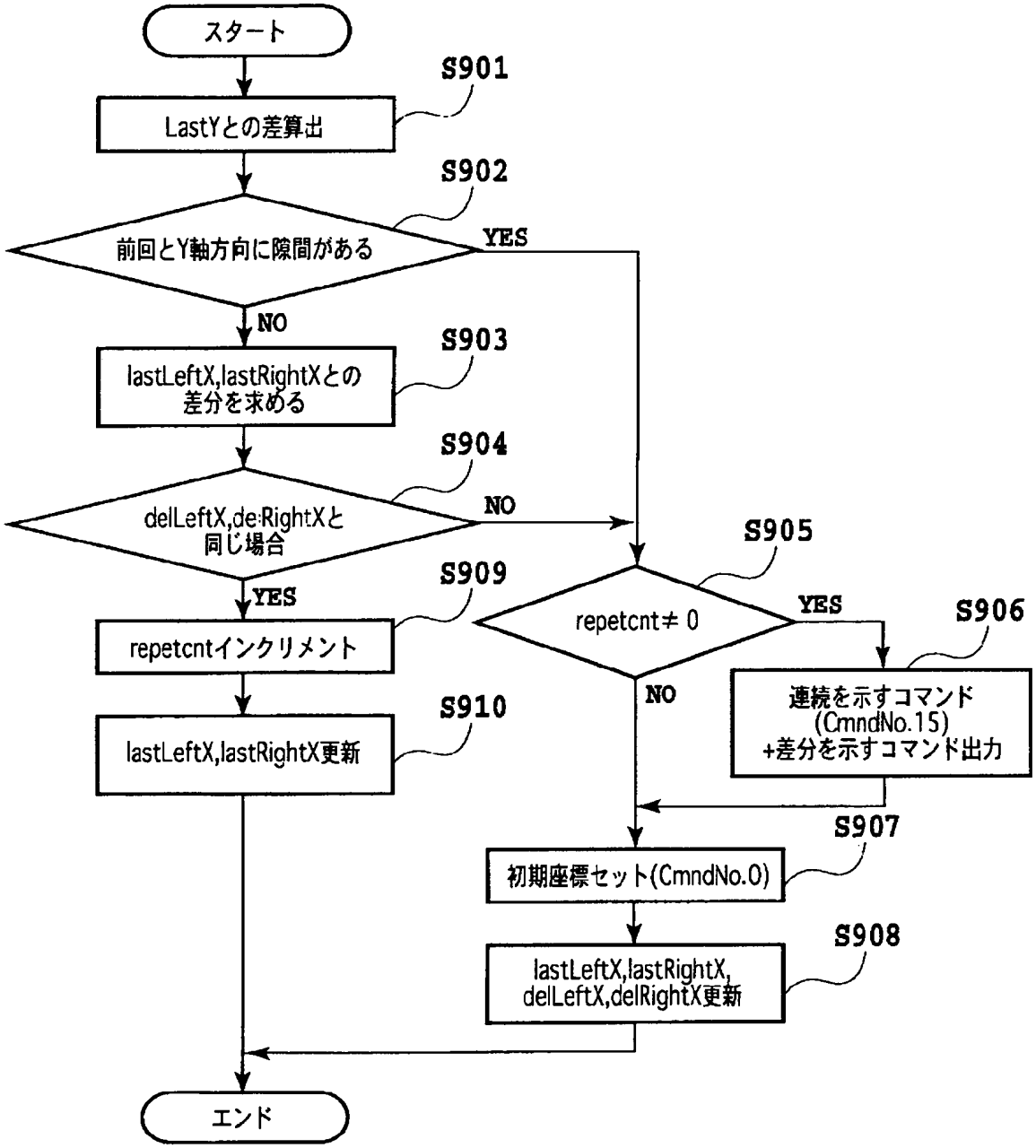
LPPDEVICE	lpDestDev	PDEVICE or PBITMAP
WORD	wStyle	OS_SCANLINES
WORD	wCount	lpPoints引数の点数
LPPOINT	lpPoints	スキャンラインの始点-終点の配列
LPPPEN	lpPPen	物理ペン
LPPBRUSH	lpPBrush	物理ブラシ
LPDRAWMODE	lpDrawMode	描画コンテキスト
LPRECT	lpClipRect	クリップ矩形



【図８】

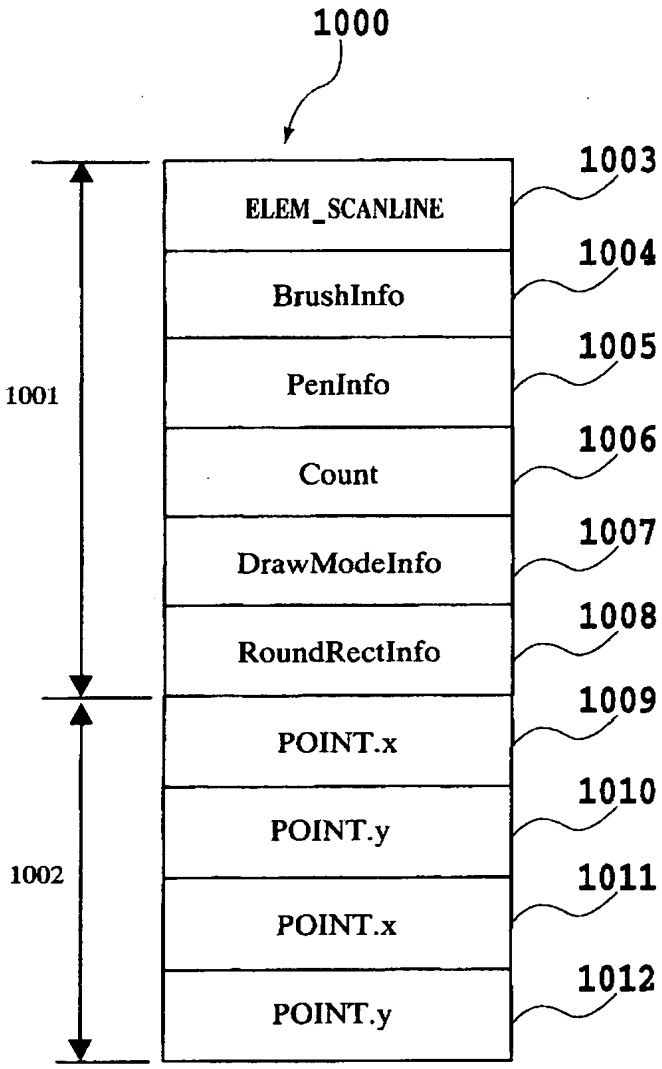

803
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
ELEM_SCANLINE
BrushInfo
PenInfo
Count
ClipInfo
ClipInfo
POINT.x
POINT.y
POINT.x
POINT.y
800
PDF生成
(0,0)
801
Output(scanline)
Output(scanline)
Output(scanline)
Output(scanline)
802
一次バッファ上で
作成の後PDF作成
804
ELEM_COMPSCANLINE
BrushInfo
PenInfo
Count
ClipInfo
ClipInfo
CompScanlineData
CompScanlineData
CompScanlineData
CompScanlineData

【図９】


スタート
S901
LastYとの差算出
S902
前回とY軸方向に隙間がある
YES
NO
S903
lastLeftX,lastRightXとの
差分を求める
S904
delLeftX,de:RightXと
同じ場合
NO
YES
S905
repetcnt≠ 0
YES
NO
S906
連続を示すコマンド
(CmndNo.15)
+差分を示すコマンド出力
S909
repetcntインクリメント
S910
lastLeftX,lastRightX更新
S907
初期座標セット(CmndNo.0)
S908
lastLeftX,lastRightX,
delLeftX,delRightX更新
エンド

【図１０】

PDFの構造を示す模式図

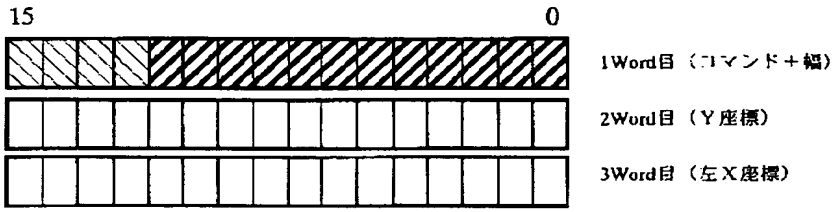


【図１１】

（コマンドデータ構成）

コマンドとは、COMPScanlineデータ部の最上位４ビット部を使い、圧縮したデータ識別用に設けられたコード（０～１２）を示し、その値により圧縮されたデータ構造が異なる。

・ｺﾏﾝﾄﾞｺｰﾄﾞ０（3Word　構成）



Ｙ座標、左Ｘ座標はサイン付パラメータを示す。

※ｻｲﾝ付とは、最上位ﾋﾞｯﾄが符号で、他のﾋﾞｯﾄは値を示す。
（ｻｲﾝﾋﾞｯﾄ以外は絶対値である）

【図１２】

・ｺﾏﾝﾄﾞｺｰﾄﾞ１、２（1Word　構成）

ｺﾏﾝﾄﾞｺｰﾄﾞ1は、前回の座標データと今回の座標データを比較し、Y座標変化量が＋１、left側座標は前回と同じ、かつright側座標変化量を11ビットで表現可能な変化範囲の場合に使用される。（ﾋﾞｯﾄ長は12だが、１ﾋﾞｯﾄは符号で使用）

同様にｺﾏﾝﾄﾞｺｰﾄﾞ2は、Y座標変化量が＋１、right側座標が前回と同じ、かつleft側座標変化量を11ビットで表現可能な場合に使われる。

【図１３】

・ｺﾏﾝﾄﾞｺｰﾄﾞ３（１Word　構成）

上記ｺﾏﾝﾄﾞｺｰﾄﾞ3は、前回の座標データと今回の座標データを比較し、Y座標変化量が＋１、leftおよびRight座標変化が5ビットで表現可能な変化範囲の場合に使用される。
（ﾋﾞｯﾄ長は6だが、１ﾋﾞｯﾄは符号で使用）

・ｺﾏﾝﾄﾞｺｰﾄﾞ４、５（１Word　構成）

ｺﾏﾝﾄﾞｺｰﾄﾞ1は、前回の座標データと今回の座標データを比較し、Y座標変化量が－１、left側座標は前回と同じ、かつright側座標変化量を11ビットで表現可能な変化範囲の場合に使用される。（ﾋﾞｯﾄ長は12だが、１ﾋﾞｯﾄは符号で使用）
同様にｺﾏﾝﾄﾞｺｰﾄﾞ2は、Y座標変化量が＋１、right側座標が前回と同じ、かつleft側座標変化量を11ビットで表現可能な場合に使われる。

【図１４】

・ｺﾏﾝﾄﾞｺｰﾄﾞ６（１Word　構成）

上記ｺﾏﾝﾄﾞｺｰﾄﾞ3は、前回の座標データと今回の座標データを比較し、Y座標変化量が－１、leftおよびRight座標変化が5ビットで表現可能な変化範囲の場合に使用される。
（ﾋﾞｯﾄ長は6だが、１ﾋﾞｯﾄは符号で使用）

・ｺﾏﾝﾄﾞｺｰﾄﾞ１５（１Word　構成）

上記ｺﾏﾝﾄﾞｺｰﾄﾞ１５は、前回と今回の座標データを比較することで求められるコマンドコード番号が前回のものと同じ場合、その連続数を12ビットで表現できるときに使用される。尚、12ビット以上になるときは、コマンド０を使用して初期座標の再設定を行う。

【書類名】　　要約書

【要約】

【課題】　プリンタドライバで中間言語を一旦スプールする場合、保持データサイズ・個数を抑えることにより高速な印刷を可能にする。

【解決手段】　プリンタドライバがＤＤＩ関数を介して受け取った描画命令の情報を、描画オブジェクトタイプ毎のタグをつけて、描画位置情報、色情報、パターン情報、描画論理情報といった属性と共に、一次バッファ３０２４に保持する。一次バッファ３０２４に保持する際に、既に保持されている描画オブジェクトがある場合、タイプ、描画属性が同じ時は一次バッファ３０２４の中に追加登録していく。「タイプが異なる。」「一次バッファにデータが入りきらない」、「描画属性が変化する」、「ページ処理終了」の何れかに当てはまるときは、既に一次バッファ３０２４に入っている描画オブジェクトからＰＤＦ３０３１（ページデータファイル）を生成し、スプールする。

【選択図】　　図３

出　願　人　履　歴　情　報

識別番号　　〔０００００１００７〕

１．変更年月日　　１９９０年　８月３０日
　［変更理由］　新規登録
　　住　所　　東京都大田区下丸子３丁目３０番２号
　　氏　名　　キヤノン株式会社